API Security @ T-Mobile

T-Mobile API Access Process (TAAP) and Zero-Trust

Today's Objectives

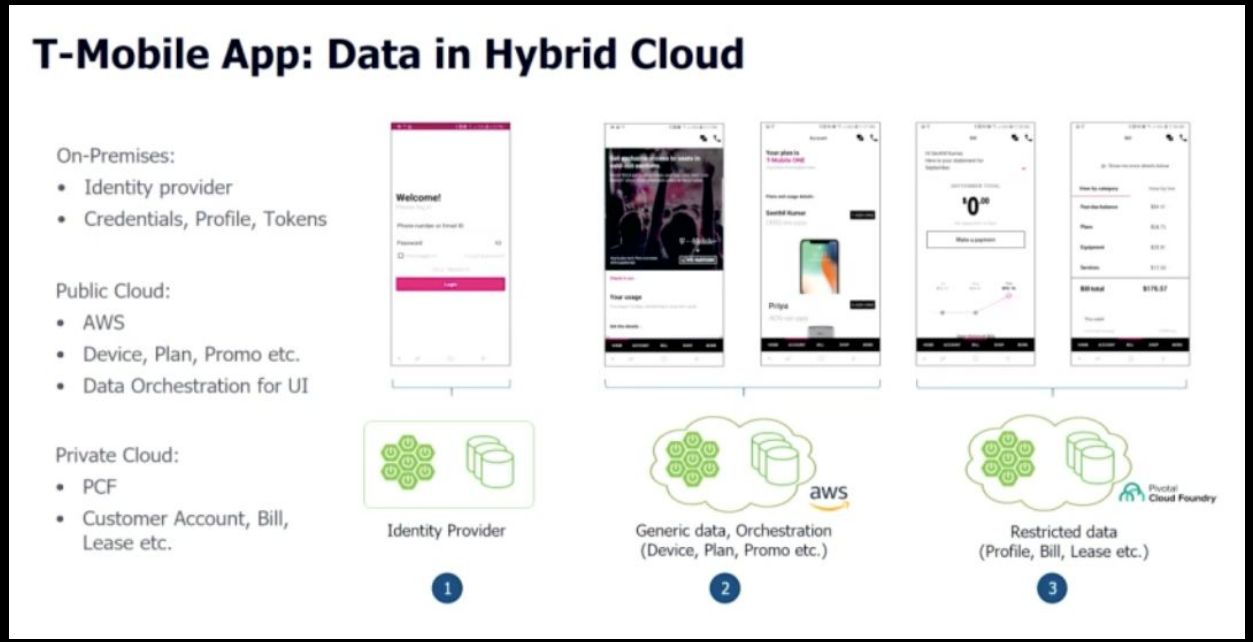
Describe T-Mobile's API Access Process (TAAP) in relation to Zero-Trust philosophy.

Work with the TAAP library in the context of securing a simple Java Spring Boot application.

Why Should You Care?

- Zero-Trust is a company-wide initiative and mindset
- TAAP is an API Security standard at T-Mobile
- TAAP is a practical application of Zero-Trust at T-Mobile
- You will need to interact with or implement TAAP-secured µ-services at T-Mobile

Many Places for Data



Source: https://springone.io/2018/sessions/securing-microservices-in-hybrid-cloud

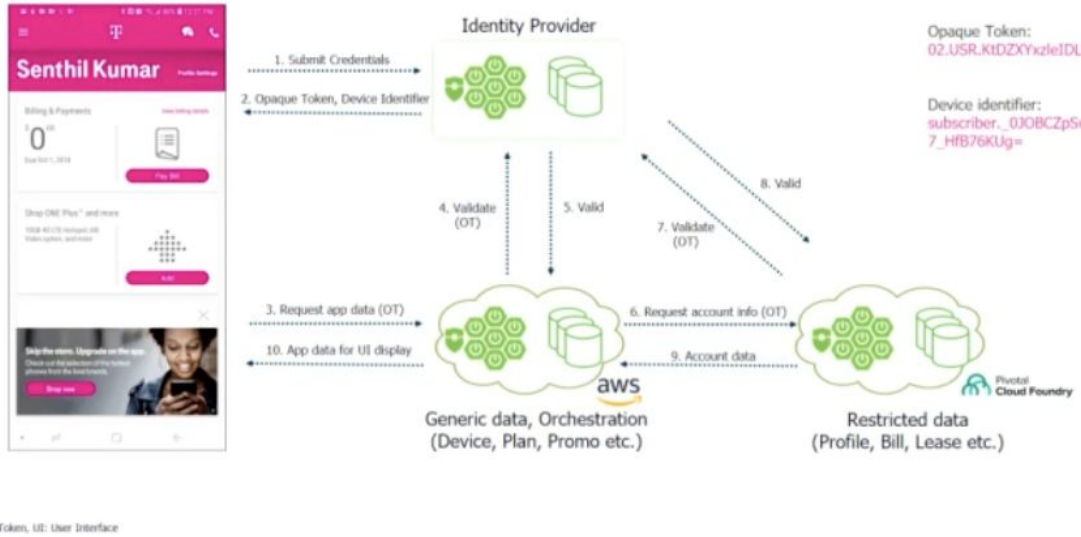
Authors: Senthil Velusamy (Sr MTS Domain Architecture, Director, T-Mobile) and Komes Subramaniam (Principal Software Engineer, T-Mobile)

A Complex Environment

- **Hybrid cloud**
  - Private cloud
  - Public cloud
  - On-premises
- **Complex AuthN/AuthZ challenges**

T-Mobile App v1

T-Mobile App (V1) : Opaque Token Call-flow



OT: Opaque Token, UI: User Interface

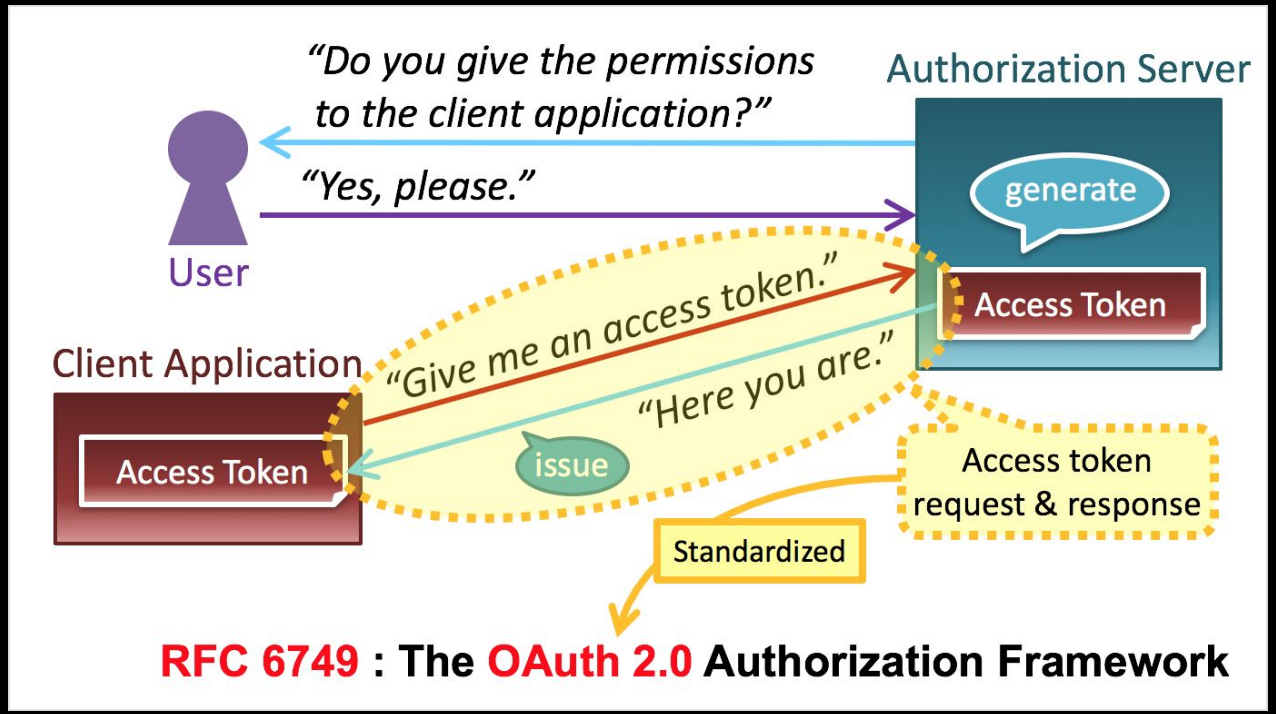
Source: https://springone.io/2018/sessions/securing-microservices-in-hybrid-cloud

Authors: Senthil Velusamy (Sr MTS Domain Architecture, Director, T-Mobile) and Komes Subramaniam (Principal Software Engineer, T-Mobile)

A Security Layer Cake

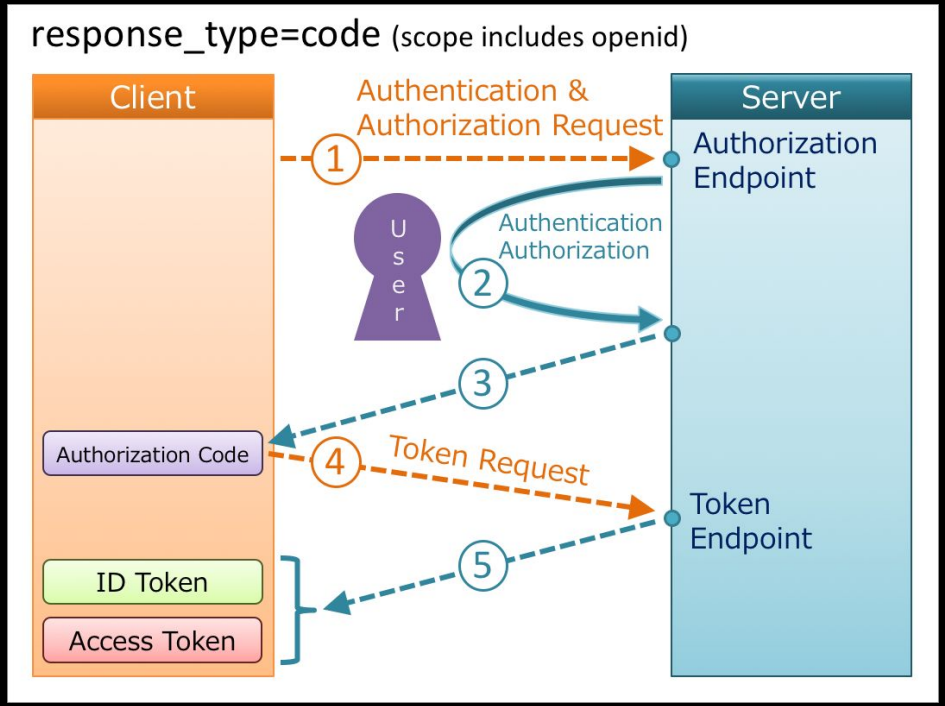
PoP	OAuth Access Token Security Enhancement
Open ID Connect (OIDC)	AuthN
OAuth 2.0	AuthZ

OAuth 2.0 - AuthZ (Access) Tokens



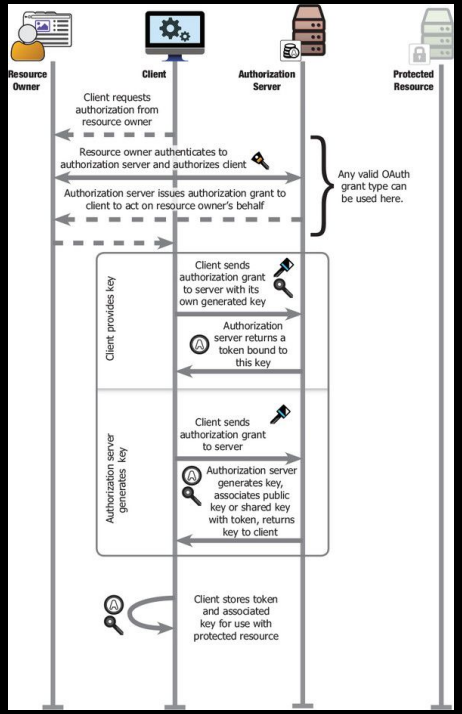
Source: https://medium.com/@darutk/the-simplest-guide-to-oauth-2-0-8c71bd9a15bb
Authors: Takahiko Kawasaki

Open ID Connect - AuthN (ID) Tokens



Source: https://medium.com/@darutk/diagrams-of-all-the-openid-connect-flows-6968e3990660
Authors: Takahiko Kawasaki

PoP: Proof of Possession Token

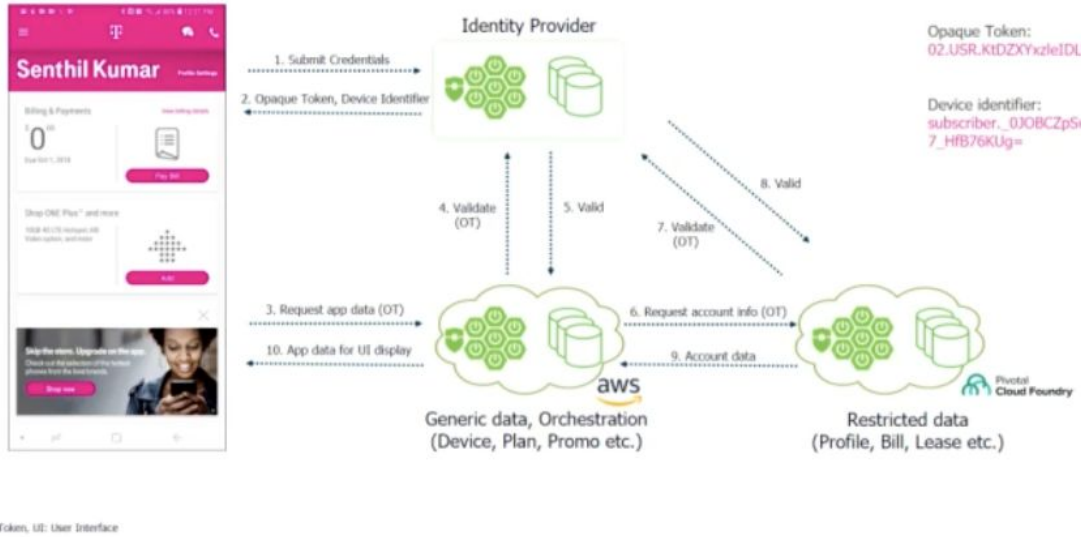


- Token theft prevention
- Client proof via public/private keys
- An additional layer of security
- Application layer
- Issuance via web-based flow

Source: OAuth 2 in Action book https://livebook.manning.com/book/oauth-2-in-action/chapter-15/82
Authors: Justin Richer, Antonio Sanso

T-Mobile App V1 to V2

T-Mobile App (V1) : Opaque Token Call-flow



OT: Opaque Token, UI: User Interface

Source: https://springone.io/2018/sessions/securing-microservices-in-hybrid-cloud

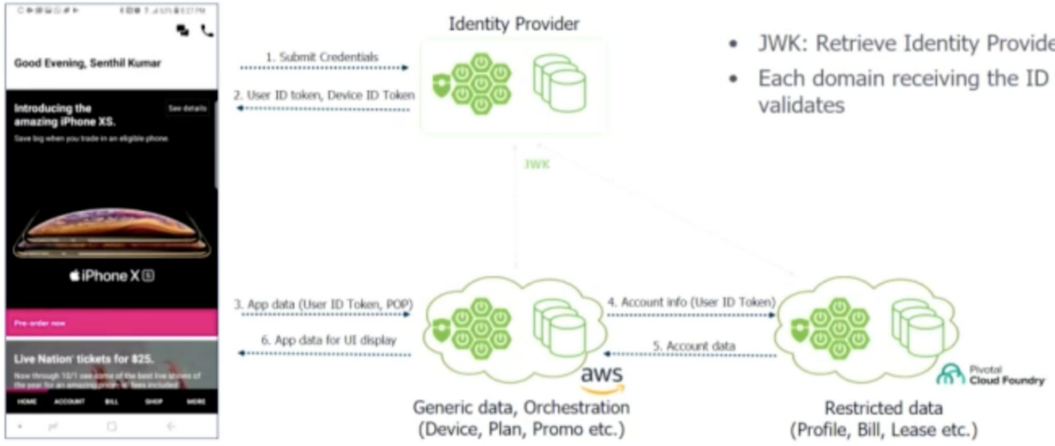
Authors: Senthil Velusamy (Sr MTS Domain Architecture, Director, T-Mobile) and Komes Subramaniam (Principal Software Engineer, T-Mobile)

T-Mobile API Environment, Revisited

- **Goals:**
  - Improve performance, latency, and scalability in a μ-service environment
  - Reduce dependency on a centralized IdP
  - Make tokens transparent and easily verifiable by μ-services
  - Enhance security via Proof of Possession of tokens for certain use cases

Enter TAAP

T-Mobile App (V2) : TAAP Call-flow



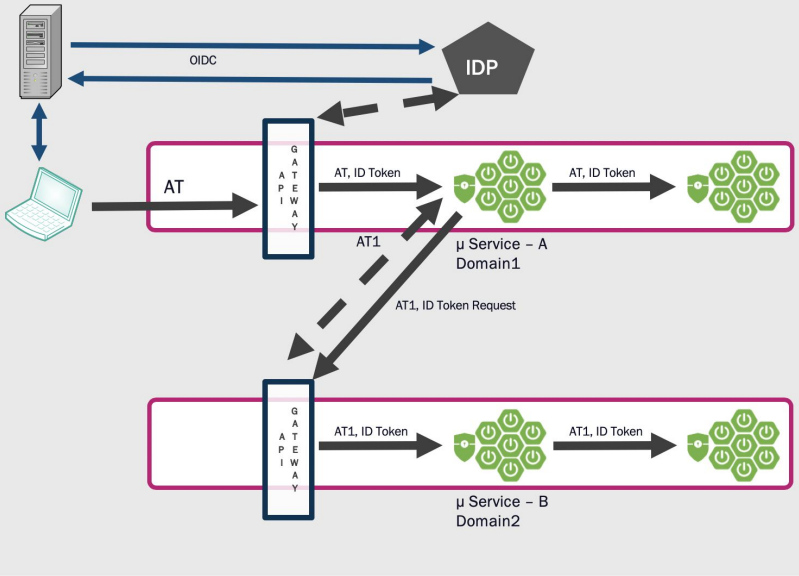
Source: https://springone.io/2018/sessions/securing-microservices-in-hybrid-cloud

Authors: Senthil Velusamy (Principal Software Engineer, T-Mobile) and Komes Subramaniam (Sr MTS Domain Architecture, Director, T-Mobile)

A TAAP API Call

TAAP: API Call Using TAAP

1. Client Application follows TAAP Flow for obtaining Access Token & ID Token
2. Client sends AT to API Gateway. Gateway does a cache lookup for ID Token
3. API Gateway sends AT & ID Tokens as part of μ Service request
4. μ Service-A may require system level access for μ Service-B. In this case, it follows Client Credential grant flow to obtain Access Token (AT1)
5. μ Service-A sends AT1 and ID Token (Original) to μ Service-B



Source: https://devcenter.t-mobile.com/documents/5ea1ee53f86d535a89d57ac4/5ea1ee53f86d535a89d57ac3?name=API-Security&sectionName=4.0-Implementing-TAAP

Improvements

- No dependence on the IdP by the µ-services
  - µ-services validate the tokens themselves
- Tokens are now JWT (JSON Web Tokens) - transparent
- Performance improvements (in this case 20% improvement)
- PoP tokens “guarantee” message integrity

Zero-Trust

- At each domain boundary
- On each request
- Device identity
- User/client identity
- Highly specific AuthZ

Checkpoint